

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平7-93010

(43)公開日 平成7年(1995)4月7日

(51)Int.Cl. ⁴	識別記号	庁内整理番号	F I	技術表示箇所
G 0 5 B 19/18				
B 2 5 J 9/10				
9/22	Z	9064-3H	G 0 5 B 19/ 18	D
		9064-3H		E
審査請求 未請求 請求項の数1 F D (全 14 頁) 最終頁に続く				

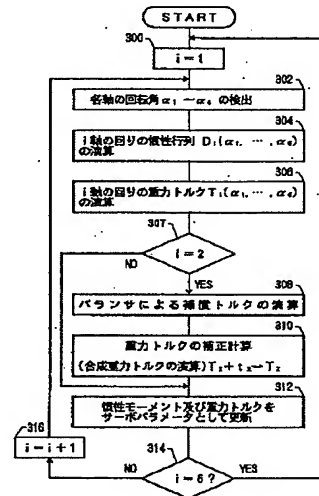
(21)出願番号	特願平5-259219	(71)出願人	000003470 豊田工機株式会社 愛知県刈谷市朝日町1丁目1番地
(22)出願日	平成5年(1993)9月22日	(72)発明者	下越 昭 愛知県刈谷市朝日町1丁目1番地 豊田工機株式会社内
		(72)発明者	吉見 光二 愛知県刈谷市朝日町1丁目1番地 豊田工機株式会社内
		(72)発明者	高野 寿男 愛知県刈谷市朝日町1丁目1番地 豊田工機株式会社内
		(74)代理人	弁理士 藤谷 修

(54)【発明の名称】 ロボットの制御装置

(57)【要約】

【目的】 バランサを有したロボットにおける重力トルクを考慮して制御性能を向上させる。

【構成】 教示点データ間の補間演算の間において、リアルタイムで、各軸の回りの慣性モーメント及び重力トルクが、角軸の現在角度及びリンクの長さ、質量等の値を用いて演算される(302-306)。又、バランサによる重力トルクを補償している軸については、角軸の現在角度、バランサの幾何学的構成やバネ定数を用いて、その時のその軸の回りの補償トルクが演算され(308)、その軸の回りの重力トルクに補償トルクが加算補正されて(310)、合成重力トルクが演算される。そして、これらの慣性モーメント及び重力トルクに応じて、速度ループのゲインや各種のフィードフォワード値が補正される。



【特許請求の範囲】

【請求項1】 軸にかかる重力トルクを減少させるためのバランスを有したロボットの制御装置であって、位置フィードバックループ、速度フィードバックループ及び電流フィードバックループを有し、前記ロボットの工具の位置及び姿勢を指令された位置及び姿勢に追従させる制御装置において、

複数の教示点における位置及び姿勢を規定した教示データを記憶する教示データ記憶手段と、

少なくとも2つの教示点間を結ぶ経路上の補間点における位置及び姿勢を規定した補間データを、少なくとも2つの教示データから補間周期毎に補間演算により求める補間演算手段と、

前記補間演算手段により演算された補間データに従って、ロボットの各軸の回転角の制御を行う角度制御手段と、

ロボットの各軸の回転角を検出する角度検出手段と、検出された各軸の回転角とロボットの各リンクの長さ及び質量とから、順次、各軸の回りの慣性モーメント及び重力トルクの演算を分割し、前記補間周期において前記補間演算手段による補間演算が行われていない空時間を用いて、数回の補間周期で慣性モーメント及び重力トルクを演算することで、各軸の回りの慣性モーメント及び重力トルクを順次更新するモーメント／トルク演算手段と、

検出された各軸の回転角、及び、バランスの構造に基づいた定数に基づいて、前記バランスの取り付けられている軸の回りの前記補償トルクによる補償トルクの演算を分割し、前記補間周期において前記補間演算手段による補間演算が行われていない空時間を用いて、数回の補間周期で、前記バランスの補償トルクを演算する補償トルク演算手段と、

前記モーメント／トルク演算手段により演算された前記重力トルクと前記補償トルク演算手段により演算された前記補償トルクとを合成するトルク合成手段と、

前記モーメント／トルク演算手段により演算され更新されているその時の慣性モーメントに基づいて前記速度フィードバックループのゲインを変化させる第1補正手段と、

前記トルク合成手段により演算され更新されているその時の合成重力トルクに基づいて前記電流フィードバックループへの指令値を加算補正する第2補正手段とから成るロボットの制御装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、軸の回りの重力トルクを補償するためのバランスを有したロボットにおいて、各軸の回りの慣性モーメント及び重力トルクがロボットの姿勢の変化に伴って変化することによって生じる軌跡制御性能の低下を防止した装置に関する。

【0002】

【従来の技術】 多関節ロボットにおいては、軸の回りの重力トルクを補償するためにバランスを取り付けたものが存在する。このバランスを取り付けたロボットにおいても、姿勢変化に伴って、各軸の回りの慣性モーメント及び重力トルクが変化する。このため、位置及び速度フィードバックループのゲインを一定とすると、ロボットの全作動領域において位置決め誤差や位置決め速度を一定にすることは困難である。例えば、位置や速度のフィードバックループのゲインを慣性モーメントや重力トルクが最大となる姿勢の時に所定の位置決め誤差及び位置決め速度が得られるように設定した場合には、慣性モーメントや重力トルクが小さくなる姿勢では、フィードバックループのゲインが高くなり過ぎ、振動が発生し、位置決め時間が長くなるという問題がある。逆に、フィードバックループのゲインを慣性モーメントや重力トルクが最小となる姿勢の時に所定の位置決め誤差及び位置決め速度が得られるように設定した場合には、慣性モーメントや重力トルクが大きくなる姿勢では、フィードバックループのゲインが不足し、位置決め誤差が増大したり、追従遅れが増大し、位置決め時間が長くなるという問題がある。

【0003】 この課題を解決するために、教示データを補間して各軸の指令角度を演算すると共にその指令角度における各軸の回りの慣性モーメント及び重力トルクを演算して、それらの値に基づいて位置及び速度フィードバックループのゲインを変化させることが考えられる。しかしながら、教示データの補間には多大な時間がかかる上に、各補間点における慣性モーメント及び重力トルクを演算するとすると、CPUの演算時間が長くなり、教示データの補間周期が長くなる。この結果、ロボットの姿勢変化速度を大きくすることができないという問題が発生する。

【0004】 又、慣性モーメント及び重力トルクの演算量を減少させる方法として、特開昭59-220806号公報に記載のものが存在する。この方法は、位置及び姿勢の教示時に、教示点での慣性モーメント及び重力トルクを演算して、教示データと共に記憶しておき、ロボットの動作時には、教示点間の補間演算と共に慣性モーメント及び重力トルクも補間演算するという方法である。この方法によれば、教示点での慣性モーメント及び重力トルクを各軸の角度から演算する必要がないことから、CPUの演算時間は短縮され、教示データの補間周期が長くないという利点がある。

【0005】

【発明が解決しようとする課題】 しかしながら、上記のように、教示時に教示点における慣性モーメント及び重力トルクを演算するという方法は、教示時において、ロボットの手首部先端に取り付けられるハンドやこのハンドが把持する工具の重心や質量が既知であることが必要

となるため、工具を動作時と同一工具にしておかなければならないという問題がある。又、別の工具を用いて同一教示データで加工する場合には、各教示点における慣性モーメント及び重力トルクが異なるため精度の高い制御を行うことが困難である。さらに、異なる重量の物体をロボットにより把持して同一経路で移動させる様な場合には、重量が異なれば、同一姿勢でも慣性モーメント及び重力トルクが異なるので、それぞれの教示データを作成する必要がある。又、重量物の把持前と把持後では、同一姿勢でも慣性モーメント及び重力トルクが異なるため、単一の教示データでは対応できないという問題がある。

【0006】又、軸の回りの重力トルクを補償するためのバランスを用いたロボットが存在するが、このバランスによる重力トルクの補償量は、ロボットの姿勢によって変化する。従って、軸の回りの重力トルクを演算する場合にも、補償量を考慮に入れる必要があり、しかも、姿勢によって補償量が変わることも考慮に入れなければ、サーボ制御性能が向上しない。

【0007】本発明は上記の課題を解決するために成されたものであり、その目的は、動作時において教示データの補間周期を長くすることがなく、バランスによる重力トルクの補償を考慮に入れて、各軸の重力トルクを演算した上で、そのトルクに応じた制御を行うことで、制御性能を向上させることである。

【0008】

【課題を解決するための手段】上記課題を解決するための発明の構成は、軸にかかる重力トルクを減少させるためのバランスを有したロボットの制御装置であって、位置フィードバックループ、速度フィードバックループ及び電流フィードバックループを有し、ロボットの工具の位置及び姿勢を指令された位置及び姿勢に追従させる制御装置において、少なくとも2つの教示点間を結ぶ経路上の補間点における位置及び姿勢を規定した補間データを、少なくとも2つの教示データから補間周期毎に補間演算により求める補間演算手段と、補間演算手段により演算された補間データに従って、ロボットの各軸の回転角の制御を行う角度制御手段と、ロボットの各軸の回転角を検出する角度検出手段と、検出された各軸の回転角とロボットの各リンクの長さ及び質量とから、順次、各軸の回りの慣性モーメント及び重力トルクの演算を分割し、補間周期において補間演算手段による補間演算が行われていない空時間を用いて、数回の補間周期で慣性モーメント及び重力トルクを演算することで、各軸の回りの慣性モーメント及び重力トルクを順次更新するモーメント／トルク演算手段と、検出された各軸の回転角、及び、バランスの構造に基づいた定数に基づいて、バランスの取り付けられている軸の回りのバランスによる補償トルクの演算を分割し、補間周期において補間演算手段による補間演算が行われていない空時間を用いて、数回

の補間周期で、バランスの補償トルクを演算する補償トルク演算手段と、モーメント／トルク演算手段により演算された重力トルクと補償トルク演算手段により演算された補償トルクとを合成するトルク合成手段と、モーメント／トルク演算手段により演算され更新されているその時の慣性モーメントに基づいて速度フィードバックループのゲインを変化させる第1補正手段と、トルク合成手段により演算され更新されているその時の合成重力トルクに基づいて電流フィードバックループへの指令値を加算補正する第2補正手段とを設けたことである。

【0009】

【作用】ロボットの動作時には、動作プログラムに従って指定の教示点への指定された移動が行われる。この時、教示点間における各軸の角度の指令値は、教示データの補間によってある補間周期毎に演算される。

【0010】この補間周期から補間演算手段による補間演算に使用される時間を引いた残りの空時間を利用して、検出された各軸の回転角とロボットの各リンクの長さ及び質量とから、順次、各軸の回りの慣性モーメント及び重力トルクが繰り返し演算される。この順次繰り返される演算により、その時の各軸の回りの慣性モーメント及び重力トルクが順次更新される。又、同様に補間周期から補間演算手段による補間演算に使用される時間を引いた残りの空時間を利用して、検出された各軸の回転角、バランスの取り付け構造及びバネ定数等に基づいて、バランスの取り付けられている軸の回りのバランスによる補償トルクが演算され、モーメント／トルク演算手段により演算された重力トルクと補償トルク演算手段により演算された補償トルクとが合成される。そして、順次、更新されているその時の各軸の慣性モーメントに基づいて速度フィードバックループのゲインが変化する。又、順次、更新されているその時の合成重力トルクに基づいて電流フィードバックループへの指令値が加算補正される。

【0011】

【発明の効果】このように、モーメント／トルク演算手段による演算は、補間周期から補間演算手段による補間演算に使用される時間を引いた残りの空時間に行われ、数回の補間周期を使って演算が完了されるので、慣性モーメント及び重力トルクの演算のために補間演算の周期が長くなるということが防止される。つまり、慣性モーメント及び重力トルクの演算は上記の残り時間を利用して分割して演算される。このような演算を行うことによって、各補間点毎の慣性モーメント及び重力トルクを演算することはできないが、リアルタイムによる演算が可能となるため、ロボット等が各種のワークを把持する場合でも容易に対応できる。又、バランスによる軸回りの重力トルクの補償量も考慮した合成重力トルクを演算しているので、姿勢に係わらず制御性能が向上する。

【0012】

【実施例】以下本発明を具体的な実施例に基づいて説明する。図1は6軸多関節ロボットの機構を示した機構図である。10がロボット本体であり、フロアに本体10を固定するベース13が配設され、ベース13上にはコラム12が図設されており、コラム12はボディ14を回転自在に配設している。ボディ14はアッパーアーム15を回転自在に軸支し、アッパーアーム15は、フォアアーム16を回転自在に軸支している。ボディ14、アッパーアーム15、フォアアーム16は、それぞれ、サーボモータSm1、Sm2、Sm3（図2参照）によって、軸a、b、cの回りに回転駆動される。この回転角はエンコーダE1、E2、E3によって検出される。フォアアーム16の先端部にはツイストリスト17がd軸の周りに回転可能に軸支され、ツイストリスト17にはベンドリスト9がe軸の周りに回転自在に軸支されている。ベンドリスト9には先端にフランジ18aを有するスイベルリスト18がf軸の回りに回転可能に軸支されている。また、フランジ18aにはハンド19が取り付けられている。d軸、e軸、f軸はサーボモータSm4、Sm5、Sm6によって駆動され、その回転角はエンコーダE4、E5、E6によって検出される。ハンド19の開閉動作は工具駆動回路23により制御される。

【0013】軸bの回りの重力トルクはロボットの姿勢によって大きく変化する。この重力トルクの姿勢による変動を補償するためにコイルスプリングによるバランサ11が設けられている。このバランサ11の一点111はボディ14において軸bを通る鉛直線上に固定されている。又、他点112はアッパーアーム15においてb軸とc軸を含む平面上であってb軸とc軸とを結ぶ線分g上に固定されている。そして、アッパーアーム15が鉛直線上方から鉛直線下方に向かって回転するに連れて、このバランサ11は伸びる。従って、アッパーアーム15は軸bの回りにバランサ11の引張力により鉛直上向きに回転する方向の補償トルクを受ける。この補償トルクが軸bの回りに姿勢によって変化する重力トルクをある程度補償する。

【0014】図2は、本発明のロボットの制御装置の電氣的構成を示したブロックダイヤグラムである。CPU20には、メモリ25、サーボモータを駆動するためのサーボCPU22a～22f、動作開始指令、ジョグ運転の指令、教示点の指示等を行う操作盤26が接続されている。ロボットに取付けられた各軸a～f駆動用のサーボモータSm1～Sm6は、それぞれサーボCPU22a～22fによって駆動される。

【0015】サーボCPU22a～22fのそれぞれは、CPU20から出力される各軸の角度指令値 $\theta_1 \sim \theta_6$ 、慣性モーメント D_i 、重力トルク T_i に基づいて、サーボモータSm1～Sm6の出力トルクを制御する。各駆動軸に連結されたエンコーダE1～E6の出力する検出角度 $\alpha_1 \sim \alpha_6$ はCPU20及びサーボCPU22

a～22fに入力しており、CPU20による各軸の慣性モーメント及び重力トルクの演算及びサーボCPU22a～22fによる位置フィードバック制御、速度フィードバック制御、電流フィードバック制御に用いられる。

【0016】メモリ25にはロボットを教示点データに従って動作させるためのプログラムが記憶されたPA領域とハンド19の位置と姿勢を表す教示点データを記憶するPDA領域とフランジ18aから先の重量物、即ち、ハンド19、又は、ハンド19とそれにより握持された物体を総合した物体の重心座標(a, b, c)とその質量 m_L 、ロボットの各アームの長さ重量、バランサ11の幾何学的位置関係及びバネ定数等の慣性モーメント、質量を記憶するSDA領域とサーボパラメータを記憶するITA領域と補間演算により求められた補間点における各軸の角度指令値 $\theta_1 \sim \theta_6$ を記憶するINA領域とエンコーダE1～E6から出力された検出角度 $\alpha_1 \sim \alpha_6$ を記憶するANG領域とが形成されている。

【0017】各軸のサーボCPU22a～22fは、図3に示す回路の機能をシーケンシャルなデジタル制御により達成するものである。即ち、各回転角の位置のフィードバックループと速度のフィードバックループとで電流のフィードバックループと速度のフィードフォワードループと加速度のフィードフォワードループとを有している。

【0018】次に、本装置の作動について説明する。図5はRAM25のPA領域に記憶されている動作プログラムである。行番号10、50、90が重量物の重心座標r(a, b, c)とその質量 m_L とを設定するための命令語である。この命令語の実行により重心座標r(a, b, c)と質量 m_L のデータがRAM25のSDA領域に記憶される。命令語の領域Aが重心座標rを表し、領域Bが質量 m_L を表す。重心座標(a, b, c)はスイベルリスト18のフランジ18aの中心に固定されたリンク座標系から見た座標、即ち、ノーマル、オリエント、アプローチの各座標(mm単位)で与えられる。又、質量はkg単位で表記される。行番号10、90は、フランジ18aから先の重量物がハンド19だけの場合のデータ設定に関する命令語であり、行番号50はハンド19により物体が握持された場合のデータ設定に関する命令語である。又、行番号20及び80のHANDOFF命令語は、ハンド19を開く命令語であり、行番号40のHANDON命令語はハンド19を閉じる命令語である。さらに、行番号30、60、70、100、110のMOVE命令語は指定された教示点Pnに移動させるための命令語である。

【0019】図5の動作プログラムにより、ロボットは、ハンド19を開いて、P1点に移動し、その後、ハンド19を閉じて物体を握持し、P2点を經由してP3点に移動して、ハンド19を開いて、物体を放置し、P

18-5

18-5

4点を経由してP5点に移動することが可能である。

【0020】図4は、CPU20による動作プログラムを解説するための主プログラムのフローチャートである。ステップ100において、MOVE命令語が解説されると、ステップ102において、現在位置から指定された教示点までハンド19を移動させるための補間演算が実行される。そして、補間演算により求められた各軸の角度指令値 $\theta_1 \sim \theta_6$ はサーボCPU22a~22f

に出力される。又、ステップ104において、LOAD命令語が解説されると、ステップ106において、その命令語に表記されているハンドから先の重量物の重心座標と質量のデータがRAM25のSDA領域に記憶される。又、ステップ108でHAND OFF命令語が解説されると、ステップ110において、工具駆動回路23にハンド19を開く指令を与える。又、ステップ112でHAND ON命令語が解説されると、ステップ114において、工具駆動回路23にハンド19を閉じる指令を与える。

【0021】ステップ102における教示点間の補間演算は、図6に示すフローチャートに従って実行される。この補間演算は、例えば、8ms毎の補間周期で実行される。補間演算は、良く知られたように、回転主軸法等を用いて行うことができる（例えば、特開昭62-164006号）。ステップ200において、ワールド座標系で表記された教示データとして与えられた開始点の位置及び姿勢と次の位置決め目標点の位置及び姿勢から回転主軸の方向ベクトルが演算され、ステップ202において、その回転主軸回りの回転角 θ が演算される。次に、ステップ204において、開始点の位置及び姿勢を基準とした補間角 $\Delta\theta$ が演算され、次のステップ206において、その補間角 $\Delta\theta$ を用いて開始点の位置及び姿勢を補間点における位置及び姿勢に変換するための姿勢変換行列Rが演算される。そして、ステップ208において、開始点の位置及び姿勢を表す同次座標行列に姿勢変換行列を作用させて、補間点における位置及び姿勢を表す同次座標行列が演算される。次に、ステップ210において、その補間点におけるワールド座標系で表記された位置及び姿勢の同次座標行列からジョイント座標系での値、即ち、各軸の回転角が演算され、この値はRAM25のINA領域に記憶される。

【0022】この状態で、本補間プログラムは終了し、ステップ212で、後述する優先度が低いレベルに設定された慣性モーメント及び重力トルク演算プログラムの再開処理（中断した命令語の記憶アドレスへの分岐命令及び状態レジスタ等の復元処理）が行われた後、その慣性モーメント及び重力トルク演算プログラムが継続起動される。

【0023】一方、8msec毎のタイマ割り込みが発生すると（ステップ214）、ステップ216において、優先度の低いレベルに設定された慣性モーメント及び重力

トルク演算プログラムへの割込処理（中断した命令語が記憶されているアドレスの記憶、状態レジスタ等のスタック処理等）が行われ、優先度の高い本補間処理のプログラムが実行される。そして、ステップ212、214において、既に、演算完了してRAM25のINA領域に記憶された補間点の角度指令値 $\theta_1 \sim \theta_g$ 、後述する慣

性モーメント及び重力トルク演算プログラムにより演算されRAM25のITA領域に記憶されているサーボパラメータ D_i, T_i がサーボCPU22a \sim 22fに出力される。この結果、各軸はその時の位置及び姿勢に対応した慣性モーメント及び重力トルクを補償したフィードフォワード制御によりそれらの角度指令値 $\theta_1 \sim \theta_g$ に

位置決め制御される。そして、ステップ216において、補間点が教示データとして与えられた目標位置に達したか否かが判定され、補間が終了していない場合に、次の補間点の位置及び姿勢の演算のために、ステップ206以下が実行される。一方、補間が終了した場合には、次の命令語を解読するために図4のメインプログラムに戻る。

【0024】上記の処理でステップ200 \sim 204は、メインプログラムから移行した補間演算の最初に実行され、ステップ214 \sim 212は、8ms毎のタイマ割り込みに同期して実行される。よって、次の出力サンプリングのために演算されている角度指令値 $\theta_1 \sim \theta_g$ が、8ms

毎にステップ218において出力される。そして、ステップ214 \sim 212は、およそ6msかかり、残りの2msが慣性モーメント／重力トルクの演算に使用される。この2msでは慣性モーメント／重力トルクの演算が完了しないので、1補間周期の空時間2msを断続的に使用して、即ち、数回の補間周期を使用して、約6 \sim 12msかけて、一回の演算が完了する。

【0025】次に、この優先度の低いレベルに設定されている慣性モーメント及び重力トルクの演算手順について、図7に示すフローチャートを参照して説明する。ステップ300において、ロボットの軸を特定する軸変数 i が1に初期設定され、ステップ302においてエンコーダ $E_1 \sim E_6$ から各軸の現在の回転角 $\alpha_1 \sim \alpha_6$ が検

出され、その値はRAM25のANG領域に記憶される。次に、ステップ304において、 i 軸の回りの慣性行列 $D_i (\alpha_1, \dots, \alpha_g)$ （慣性モーメントの行列表現）

が演算され、ステップ306において、 i 軸の回りの重力トルク $T_i (\alpha_1, \dots, \alpha_g)$ が演算される。尚、このス

テップでの重力トルクの演算では、 b 軸の回りの重力トルクには、バランサ12による補償トルクが加味されていない。又、 i 軸の回りのトルク τ_i は、次式で良く知られたように、ロボットアームに関するラグランジュ関数の微分から求めることができる。

【0026】

$$\begin{aligned} \text{【数1】 } \tau_i &= \sum_{j=1}^n \sum_{k=1}^j \text{Tr} (U_{jk} U_j U_{ji}^T) D_k^2 \theta_k - \\ &\quad \sum m_j g U_{ji}^T r_j \end{aligned}$$

18-8

18-8

但し、コリオリ力と遠心力の成分は小さいとして上式では除外した。 U_j は j 軸を構成する部材だけによるリンク慣性行列、 Tr は行列式の対角成分の和、 m_j は j 軸リンクの質量、 g は重力加速度、 ${}^j r_j$ は j 軸リンク座標

系で表された j 軸リンクの重心のベクトルであり、 $D^2 \theta_k$ は角度の時間に関する 2 次微分、即ち、角加速度である。

又、上式の第 1 項の展開式において、 $D^2 \theta_i$ の係数

が慣性行列 D_i となり、第 2 項が i 軸の周りの重力トルク T_i となる。又、 U_{jk} は $J_j =$

【0027】

$$J_j = \begin{bmatrix} (I_{xx} + I_{yy} + I_{zz})/2 & I_{xy} & I_{xz} & m_j r_{ix} \\ I_{xy} & (I_{xx} - I_{yy} + I_{zz})/2 & I_{yz} & m_j r_{iy} \\ I_{xz} & I_{yz} & (I_{xx} + I_{yy} - I_{zz})/2 & m_j r_{iz} \\ m_j r_{ix} & m_j r_{iy} & m_j r_{iz} & m_j \end{bmatrix}$$

【数2】 0 ($j < i$ の時)

【数3】 $\delta ({}^0 A_1 {}^1 A_2 \dots {}^{j-1} A_j) / \delta \theta_i = \Pi_{s=i}^{j-1} {}^j A_s$

$\cdot \delta^{i-1} A_i / \delta \theta_i$ ($j \geq i$ の時)

但し、 $s \neq i$ である。又、 ${}^{j-1} A_j$ は j 軸リンク座標を j

-1 軸リンク座標へ変換する同次座標変換行列であり、

${}^0 A_1$ は 1 軸リンク座標をワールド座標に変換する同次座標変換行列である。

【0028】 U_i は慣性テンソル I_{ij} を用いて次の様に

【0029】但し、 r_{ix} 、 r_{iy} 、 r_{iz} はj軸リンク座標から見たj軸リンクの重心の座標であり、 m_j はその質量である。又、

$$\text{【数5】 } I_{ij} = \int (\delta_{ij} (\sum x_k^2) - x_i y_j) dm$$

である。

【0030】尚、LOAD命令語によって設定される重心座標(a, b, c)及び質量 m_L により、 J_g が変化する。 J_g は、数4において、(r_{ix} , r_{iy} , r_{iz}) = (a, b, c)、 $m_j = m_L$ とおくことにより求めることができる。

【0031】数1から、現在の検出角度($\alpha_1, \dots, \alpha_g$)における各軸の回りの慣性行列 D_i と重力トルク T_i とが演算される。本実施例では、演算時間を短縮するために、他軸との相互作用は小さいとして、(i, i)対角成分のみを考慮している。請求の範囲の慣性モーメントの用語は、対角成分だけでなく慣性行列を含む意味で用いているが、以下の記載では、便宜上、慣性行列 D_i の(i, i)対角成分を、慣性モーメント D_i と記す。

【0032】次に、ステップ307において、 $i=2$ 、即ち、b軸であるならば、ステップ308においてにおいて、バランス11によるb軸の回りの補償トルクが演算される。補償トルクの演算方法を次に述べる。バランス11の幾何学的配置関係は図9に示すとおりである。

【0033】

$$\text{【数6】 } l = (r^2 + h^2 - 2r \times h \times \cos \alpha)^{1/2}$$

$$\text{【数7】 } \delta = l - (r - h)$$

$$\text{【数8】 } \beta = \tan^{-1} \{ h \sin \alpha / (r - h \cos \alpha) \}$$

$$\text{【数9】 } f = \delta \times k + f_0$$

$$\text{【数10】 } t_2 = r \times f \times \sin \beta \times n$$

【0034】但し、

r : アッパーアーム15の長さ (m)

l : バランス11の長さ (m)

h : バランス11の取付け高さ (m)

k : バネ定数 (N/m)

f_0 : バネ初期力 (N)

n : バランサー数

α : ジョイント角 θ_2 (rad)

β : アッパーアーム15とバランス11とのなす角 (rad)

γ : バランス11のたわみ (m)

f : バランス11本当たりの補償力 (N)

t_2 : b軸の回りのバランス11による補償トルク (N・m)

【0035】次に、ステップ310において、i軸がb軸($i=2$)の場合には、b軸に関してステップ306で得られた重力トルク T_2 にステップ308で得られた補償トルク t_2 を加算して、この値を新しい合成重力トルク T_2 とする。

【0036】ステップ312において、慣性モーメント D_i と重力トルク T_i がサーボパラメータとしてRAM

25のITA領域に更新記憶される。この値が図6のステップ220においてサーボCPU22a~22fに出力される。次に、ステップ310において、軸変数iが最大値6に達したか否かが判定され、その値に達していない場合には、ステップ312において、軸変数iが1だけ加算されて、ステップ302に戻り、次の軸に関する慣性モーメント及び重力トルクの演算が繰り返し実行される。又、軸変数iが最大値6になれば、ステップ300に戻り、軸変数iを1に初期設定して、1軸から演算が繰り返し実行される。

【0037】この図7に示すプログラムは、上述したように、図4、図6に示すプログラムよりも優先度が1ランク低く設定されているので、図4、図6に示すプログラムが実行されていない時に実行される。このようにして、補間演算手段(図6に示す手順を実行する手段)の空き時間(2ms)において、図7に示す慣性モーメント及び重力トルクの演算が実行される。従って、図7のプログラムは、連続されて実行されるのではなく、8msecの補間演算毎に中断され、中断された次の命令語から再開される。このため、補間周期とは別の遅い周期で各軸毎に慣性モーメント及び重力トルクが順次、最新の値に更新されていくことになる。

【0038】次に、 i 軸のサーボCPUの動作手順を図8のフローチャートに従って説明する。ステップ400において、エンコーダE_iから現在の検出角度 α_i が読み取られ、ステップ402において検出角度 α_i の時間変化率から角速度 β_i 、角速度 β_i の時間変化率から角加速度 γ_i が演算される。次、ステップ404において、RAM25の領域T_Aに記憶されている最新の重カトルク T_i と慣性モーメント D_i がCPU20を介して受信され、ステップ406において、RAM25の領域I_{NA}に記憶されている角度指令値 θ_i がCPU20を介して受信される。次に、ステップ408において、角加速度 γ_i と、慣性モーメント D_i とバネ定数 K_e と定数 K_2 に基づいて、次式により、角度指令値 θ_i のフィードフォワード補正が行われる。

【0039】

$$【数11】 \theta_{inew} = \theta_i + \gamma_i D_i K_2 / K_e$$

【0040】次に、ステップ410において、補正された角度指令値 θ_{inew} の検出角度 α_i に対する角度偏差 $\Delta\theta_i (= \theta_{inew} - \alpha_i)$ が演算される。次に、ステップ412において、角度偏差 $\Delta\theta_i$ 、位置ループゲイン K_p を用いて、次式により、角速度指令値 ω_i が演算される。

【0041】

$$【数12】 \omega_i = K_p \Delta\theta_i$$

【0042】次に、ステップ414において、検出された角速度 β_i 、ゲイン K_f を用いて、角速度のフィードフォワード補正が次式により行われる。

$$【数13】 \omega_{inew} = \omega_i + \beta_i K_f$$

次に、ステップ416において、補正された角速度指令値 ω_{inew} の検出された角速度 β_i に対する角速度偏差 $\Delta\omega_i (= \omega_{inew} - \beta_i)$ が演算される。

【0043】次に、ステップ418において、電流指令値 I_i が、定数 K 、慣性モーメント D_i を用いて、次式により、演算される。

$$【数14】 I_i = K D_i \Delta\omega_i$$

但し、 D_i は上記の慣性モーメント D_i に i 軸のサーボモータのロータの慣性モーメントを加算した全慣性モーメントである。このように電流指令値を求めるための速度フィードバックループのゲインが全慣性モーメント D_i により補償されることになる。即ち、慣性モーメントが大きな姿勢では、追従性を向上させるために速度フィードバックループのゲインが大きくなるように制御される。

【0044】次に、ステップ420において、角加速度 γ_i と、全慣性モーメント D_i とトルク定数 K_{tn} と定数 K_a を用いて、次式により、電流指令値がフィードフォワード補正される。

$$【数15】 I_{inew} = I_i + \gamma_i K_a D_i / K_{tn}$$

次に、ステップ422において、最新の重カトルク T_i

重カトルクによる補正値が演算される。

$$【数16】 I_{inew} = I_{inew} + T_i K_t$$

【0045】次に、ステップ424において、最終的に演算された電流指令値 I_{inew} に基づいて、 i 軸のサーボモータに対する電流制御が実行され、このサーボモータは慣性モーメント及び重カトルクの補償がされた状態で角度指令値 θ_i に向けて位置決めされる。

【0046】以上の処理が微小時間間隔で繰り返し実行されることにより、図3に示す回路機能と等価な処理が実行される。尚、上記実施例では、慣性行列 D_i の対角成分だけで慣性モーメントの補償を行ったが、他の軸との相互作用を表した非対角成分を含めた補償を行うようにしても良い。

【0047】尚、上記実施例では、床置きタイプのロボットの案件について述べたが、天吊りタイプのロボットの場合には、図9の配置関係のバランスにおいて、数9、数10式に代えて、次の数17、数18式となる。

$$【数17】 f = \delta \times k + f_0$$

$$【数18】 t_2 = r \times f \times \sin \beta \times n$$

【0048】又、上記実施例では、コイルスプリングのバランスを用いているが、図10に示すようなエアバランスを用いても良い。その場合には、補償トルクの演算式は次のようになる。

(1) エアバランスの床置きタイプの場合

$$【数19】 l = (r^2 + h^2 - 2r \times h \times \cos \alpha)^{1/2}$$

$$【数20】 \delta = l - (r - h)$$

$$【数21】 \beta = \tan^{-1} \{ h \sin \alpha / (r - h \cos \alpha) \}$$

$$【数22】 A = (\pi / 4) (D^2 - d^2)$$

$$【数23】 P_2 \times \{ V_2 + n \times A \times (s - \delta) \} = P_1 \times (V_2 + n \times A \times s)$$

$$【数24】 P_2 = \{ P_1 \times (V_2 + n \times A \times s) \} / \{ V_2 + n \times A \times (s - \delta) \}$$

$$【数25】 f = A \times P_2$$

$$【数26】 t_2 = r \times f \times \sin \beta \times n$$

【0049】ただし、

s: シリンダーの全ストローク (m)

D: シリンダーの直径 (m)

d: ピストンロッドの直径 (m)

A: シリンダーの有効断面積 (m²)

V₂: サージタンクの体積 (m³)

P₁: $\alpha = 0$ の時の圧力 (Pa)

P₂: $\alpha = \text{ジョイント角 } \theta_2$ の時の圧力 (Pa)

n: バランス数

【0050】(2) エアバランスの点吊りタイプの場合

$$【数27】 l = (r^2 + h^2 - 2r \times h \times \cos \alpha)^{1/2}$$

$$【数28】 \delta = l - (r - h)$$

$$【数29】 \beta = \tan^{-1} \{ h \sin \alpha / (r - h \cos \alpha) \}$$

$$【数30】 A = (\pi / 4) D^2$$

$$【数31】 I_L = (r^2 + h^2 - 2r \times h \times \cos \alpha_L)^{1/2}$$

【数32】 $\delta_L = l_L - (r - h)$

【数33】 $P_2 \times (V_2 + n \times A \times \delta) = P_1 \times (V_2 + n \times A \times \delta_L)$

【数34】 $P_2 = \{P_1 \times (V_2 + n \times A \times \delta_L)\} / (V_2 + n \times A \times \delta)$

【数35】 $f = -A \times P_2$

【数36】 $t_2 = r \times f \times \sin \beta \times n$

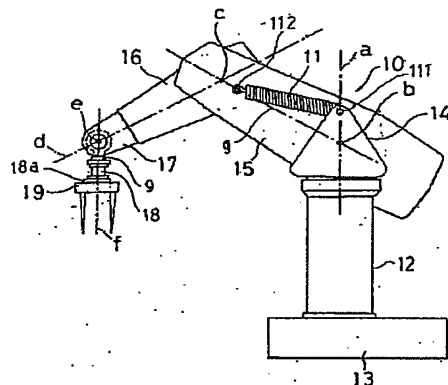
【0051】 ただし、
 l_L : $\alpha = \alpha_L$ の時のバランサーの長さ (m)

α_L : ジョイント2のリミット角 (rad)

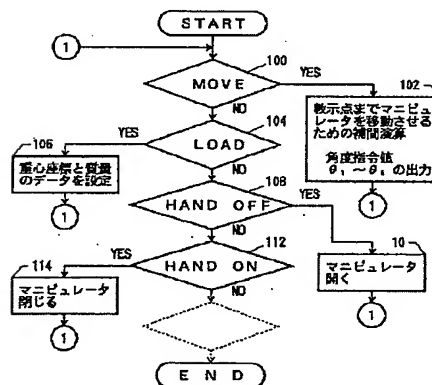
【図面の簡単な説明】
 図1は本発明の具体的な実施例装置で用いられるロボットを模式的に示した図。
 図2はロボットの制御装置の構成を示したブロック図。
 【図3】 図2におけるサーボCPUの処理を示したブロック図。
 【図4】 ロボットの位置及び姿勢制御のための主プログラムを示したフローチャート。
 【図5】 ロボットの位置及び姿勢制御のための動作プログラムを示した説明図。
 【図6】 補間演算の手順を示したフローチャート。
 【図7】 慣性モーメント／重力トルクの演算手順を示したフローチャート。

【図8】 サーボCPUによる角度制御の処理手順を示したフローチャート。
 【数9】 スプリングバランサの取付け状態における幾何学的構成を示した説明図。
 【数10】 エアーバランサの取付け状態における幾何学的構成を示した説明図。
 【符号の説明】
 10…ロボット
 11…バランサ
 15…アッパーアーム
 120…バランサ
 18…スィベルリスト
 18a…フランジ
 19…ハンド
 20…CPU (補間演算手段、モーメント／トルク演算手段、補償トルク演算手段、トルク合成手段)
 22a～22f…サーボCPU (角度制御手段、第1補正手段、第2補正手段)
 25…RAM (教示データ記憶手段)
 E1～E6…エンコーダ (角度検出手段)
 ステップ200～222…補間演算手段
 ステップ300～306…モーメント／トルク演算手段
 ステップ308…補償トルク演算手段
 ステップ310…トルク合成手段
 ステップ400～424…角度制御手段

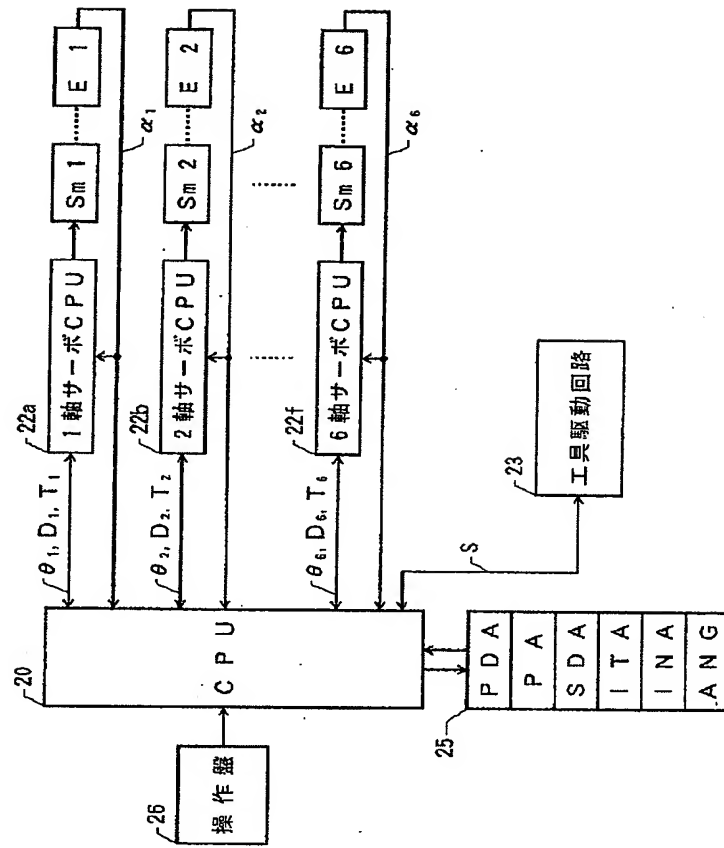
【図1】



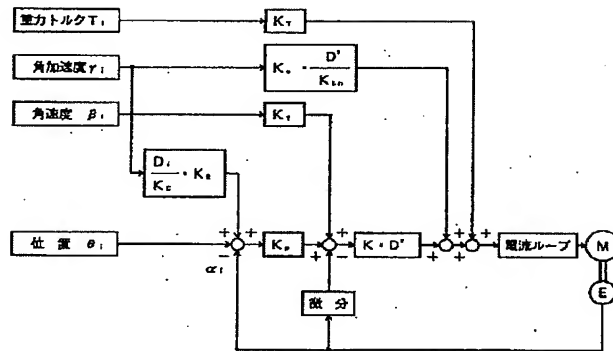
【図4】



【図2】



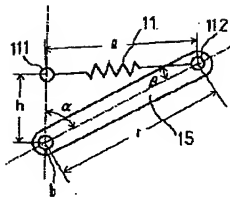
【図3】



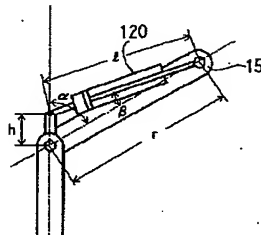
【図5】

	A	B
10	LOAD (0, 0, 150.0),	5.0
20	HAND OFF	
30	MOVE P1	
40	HAND ON	
50	LOAD (200, 150, 4000),	50.00
60	MOVE P2	
70	MOVE P3	
80	HAND OFF	
90	LOAD (0, 0, 150.0),	5.0
100	MOVE P4	
110	MOVE P5	
120	END	

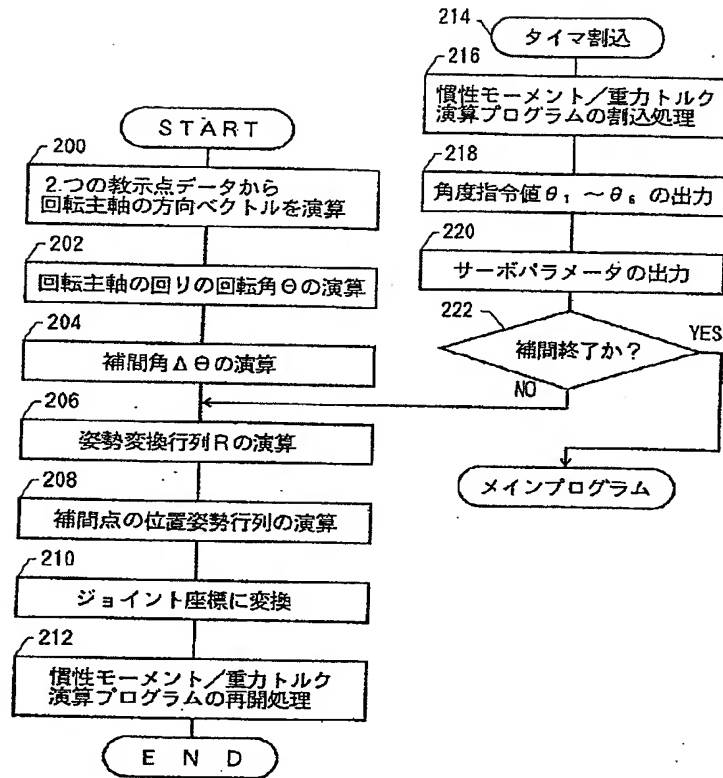
【図9】



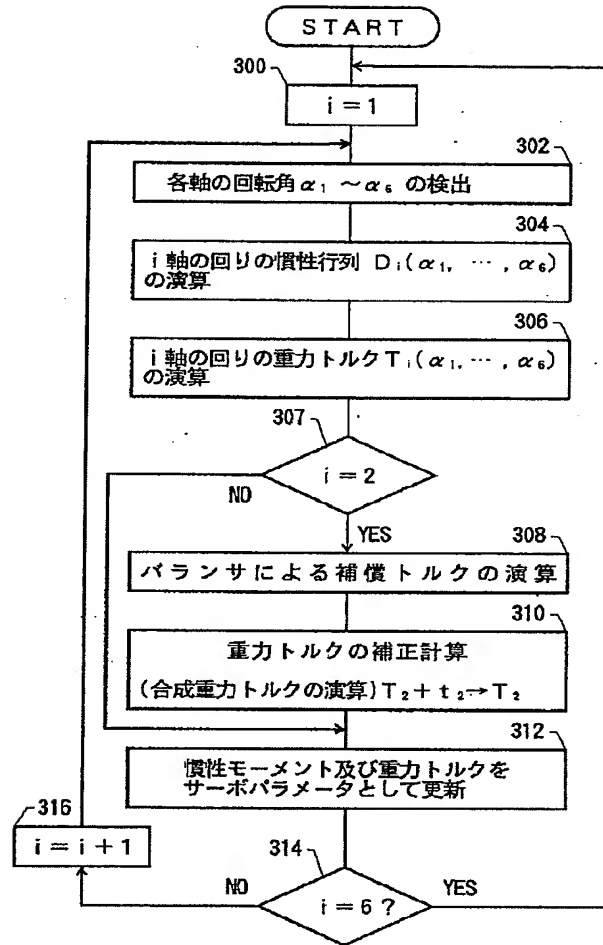
【図10】



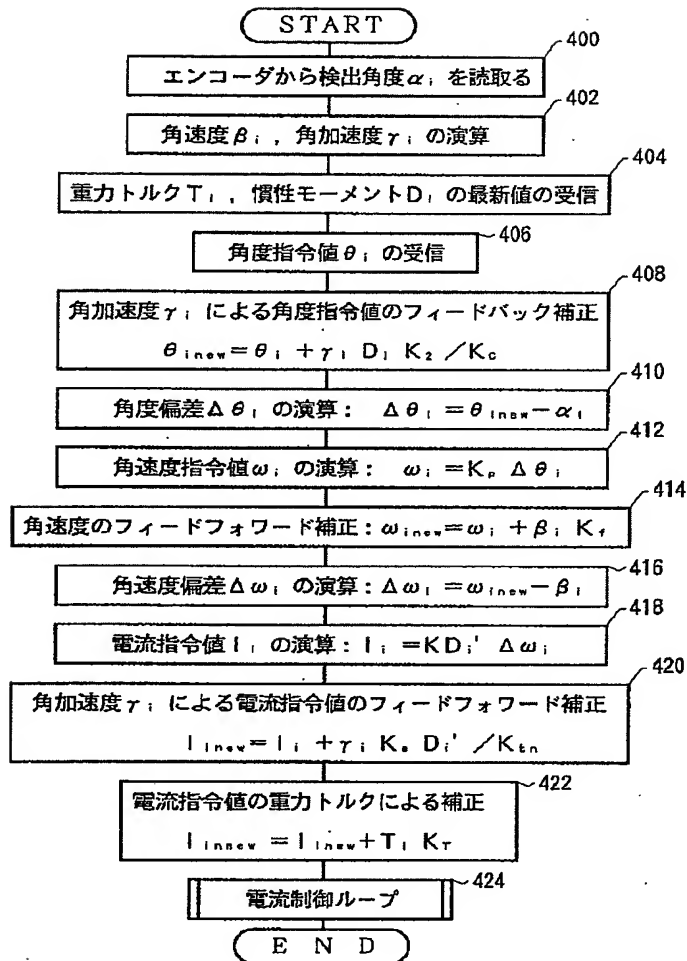
【図6】



【図 7】



【図 8】



【手続補正書】

【提出日】平成 6 年 5 月 9 日

【手続補正 1】

【補正対象書類名】明細書

【補正対象項目名】図面の簡単な説明

【補正方法】変更

【補正内容】

【図面の簡単な説明】

【図 1】本発明の具体的な実施例装置で使用するロボットを示した構成図。

【図 2】ロボットの制御装置の構成を示したブロック

- 図.
- 【図 3】図 2 におけるサーボ CPU の処理を示したブロック図。
- 【図 4】ロボットの位置及び姿勢制御のための主プログラムを示したフローチャート。
- 【図 5】ロボットの位置及び姿勢制御のための動作プログラムを示した説明図。
- 【図 6】補間演算の手順を示したフローチャート。
- 【図 7】慣性モーメント／重力トルクの演算手順を示したフローチャート。
- 【図 8】サーボ CPU による角度制御の処理手順を示したフローチャート。
- 【図 9】スプリングバランサの取付け状態における幾何学的構成を示した説明図。
- 【図 10】エアーバランサの取付け状態における幾何学的構成を示した説明図。
- 【符号の説明】
- 10…ロボット
- 11…バランサ
- 15…アッパーアーム
- 120…バランサ
- 18…スイベルリスト
- 18a…フランジ
- 19…ハンド
- 20…CPU（補間演算手段、モーメント／トルク演算手段、補償トルク演算手段、トルク合成手段）
- 22a～22f…サーボ CPU（角度制御手段、第 1 補正手段、第 2 補正手段）
- 25…RAM（教示データ記憶手段）
- E1～E6…エンコーダ（角度検出手段）
- ステップ 200～222…補間演算手段
- ステップ 300～306…モーメント／トルク演算手段
- ステップ 308…補償トルク演算手段
- ステップ 310…トルク合成手段
- ステップ 400～424…角度制御手段

フロントページの続き

(51)Int.Cl. ⁸	識別記号	庁内整理番号	F I	技術表示箇所
G 0 5 B	19/404			
	19/4103			
G 0 5 D	3/12	3 0 5 L	9179-3H	
			9064-3H	
			G 0 5 B	19/415 R